



41 AF/2177
NCR Docket No. 8779

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
Dayton, Ohio

Docket No. 8779

Application of:

Paul Sinclair, et al.

Group Art Unit: 2177

Serial No.: 09/713,887

Examiner: Mohammad Ali

Filed: November 16, 2000

For: **LOCKING DATA IN A DATABASE AFTER AN OPERATION HAS
BEGUN**

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

RECEIVED

JUL 01 2004

Technology Center 2100

BRIEF IN SUPPORT OF APPEAL

This is a brief in support of Applicant's appeal filed on April 27, 2004, in response to the final office action, dated November 27, 2003, in this matter. Applicant is filing this brief in triplicate, along with the required fee.

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to the Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313, on:

6-25-04 Sallie Spicer
Date Sallie Spicer

NCR Docket No. 8779

(1) REAL PARTY IN INTEREST

The real party in interest in this matter is NCR Corporation, Dayton, Ohio, by virtue of an assignment recorded at reel 11339, frame 0581, on November 16, 2000.

(2) RELATED APPEALS AND INTERFERENCES

Applicant is aware of no active appeals or interferences related to this application.

(3) STATUS OF CLAIMS

Claims 1 through 30 are pending. All have been rejected and are under appeal. A listing of claims is attached as an appendix to this brief.

(4) STATUS OF AMENDMENTS

All amendments have been entered prior to appeal and are reflected in the listing of claims appended to this brief.

(5) SUMMARY OF INVENTION

The invention involves a technique for use in managing data in a database system, in particular executing one or more locks on a set of target data when performing an operation on that data. As defined in claims 1, 7, 15 and 21, the technique involves receiving a request to perform an operation (*e.g.*, a data-manipulation or data-definition operation) on a set of target data residing in the database system and then executing that operation in the database system where the target data resides. At some point after execution of the operation has begun, a lock is placed on the target data to prevent concurrent execution of other operations on the target data. It is important to note that (1) the lock is not placed on the target data until after execution of the operation has begun,

NCR Docket No. 8779

and (2) the operation is executed directly on the set of target data residing in the database system.

As defined in claim 30, the technique involves receiving an instruction from a user to perform a data-definition operation (*e.g.*, a CREATE TABLE, CREATE INDEX, or DROP USER operation) on a set of target data and, in response to that instruction, placing an initial lock (*e.g.*, an ACCESS or READ lock) on the target data at a level that prevents at least one type of concurrent operation on the data. At some point after execution of the data-definition operation has begun, the lock is upgraded to a higher locking level (*e.g.*, an EXCLUSIVE lock), one that excludes all types of concurrent operations on the target data.

As defined in claim 28, the technique involves receiving an instruction to perform a particular type of data-definition operation – one that modifies the characteristics of the database itself or of a user of the database – and then initiating execution of that operation on the targeted data or user. At some point while the operation is executing, another operation is executed concurrently on objects within the same database or on the same user. The technique, as defined in this claim, thus allows execution of concurrent operations on a targeted database or user while the characteristics of that database or user are being modified.

(6) ISSUE

The issue before the Board is whether claims 1-30 are patentable under 35 U.S.C. § 102(e) in view of U.S. Patent 6,070,170, to Friske.

(7) GROUPING OF CLAIMS

Group 1: Claims 1-27 stand or fall together.

Group 2: Claims 28 and 29 stand or fall together. These claims are distinct from the Group 1 claims because they involve the concurrent execution of

NCR Docket No. 8779

operations on a targeted database or user when a particular type of data-definition operation – one that modifies the characteristics of the database itself or of the user – is being executed.

Group 3: Claim 30 stands or falls alone. This claims is distinct from the Group 1 and Group 2 claims because it involves placing an initial lock on a set of target data upon receiving an instruction to perform a data-definition operation and, at some point after the operation has begun, upgrading the lock to one at a higher locking level.

(8) ARGUMENT

A. Claims 1-27

Friske does not show, nor does he suggest, executing an operation on a set of target data “in the database” where it resides and then, “at some point after execution has begun, placing a lock on the target data to prevent concurrent execution of other operations” on that data, as claimed. Friske states clearly that a “target data set” to be reorganized “is ‘unloaded’ from the logical database” by “copying [the] data set to flat files” and then placing “the unloaded target data . . . into a shadow location” in a storage unit. (Col. 6, lines 5-11 and 25-28.) Friske then carries out the reorganization operation on this shadow copy, applying to it “log records” that reflect “changes which occurred to the original data set after the target data set was unloaded.” (Col. 6, lines 33-35.) Finally, after the target data set is updated at its shadow location, “the original data set is then replaced with the target data set.” (Col. 6, lines 42-43.)

This lengthy passage in Friske makes it clear that Friske does not execute an operation directly on a set of target data that resides in the database, but instead “unloads” the data from the database to a shadow location before operating on it. As a result, Friske does not meet the limitations of these claims, and the claims are allowable over this reference.

NCR Docket No. 8779

B. Claims 28-29

Despite the Office's assertion to the contrary on page 2 of the action dated November 28, 2003, Friske does not even hint at, let alone teach, the execution of a MODIFY DATABASE/USER operation, as claimed by Applicant. The MODIFY DATABASE/USER operation is a particular type of data-definition operation that allows modification of (1) the characteristics of a database user or (2) database options, such as the amount of permanent space ("PERM"), temporary space ("TEMP"), or spool space ("SPOOL") allocated to the database. (See Applicant's disclosure, page 7.) When applied to the database, the MODIFY DATABASE/USER operation does not modify data residing within the database, but rather modifies the structure or characteristics of the database itself.

Friske is simply oblivious to this type of data-definition operation. As such, it follows that Friske cannot possibly show or suggest the concurrent execution of other operations "within the targeted database or user" while executing such a data-definition operation. These claims, therefore, are allowable over Friske.

C. Claim 30

The Office has tried in its most recent action (pages 2-3) to equate Friske's "non-blocking drain" with both the "initial lock" and the "final lock" of Applicant's claim. This comparison, however, simply does not hold. Applicant's "initial lock" is one "that prevents at least one type of concurrent operation on the target data," while the "final lock" is one "that excludes all types of concurrent operations on the target data." Friske's "non-blocking drain" does neither.

In Friske's own words, "the non-blocking drain does not acquire a traditional lock on the target data set With the non-blocking drain, any requests to access the target data set will not be blocked This allows other

NCR Docket No. 8779

processes that need to use the target data set to access the data set even when the reorganization process is taking place.” (Col. 6, lines 58-67, emphasis added.)

Both the “initial lock” and the “final lock” of Applicant’s claim prevent “at least one type of concurrent operation on the target data.” Friske’s “non-blocking drain,” on the other hand, prevents no operations whatsoever. This “non-blocking drain,” therefore, could serve as neither the “initial lock” nor the “final lock” of Applicant’s claim, and thus the claim is patentable over the Friske reference.

(9) CONCLUSION

The Friske patent does not show or suggest the features of Applicant’s invention as set forth in any of the claims. Applicant therefore asks the Board to reverse the rejection and allow all of the claims.

Please apply any charges or credits that might be due, except the issue fee, to Deposit Account 50-1673.

Respectfully,



John D. Cowart
Reg. No. 38, 415

6/24/04

NCR Corporation
1700 South Patterson Blvd.
Dayton, Ohio 45479

Tel. No. (858) 485-4903
Fax No. (858) 485-2581

NCR Docket No. 8779

APPENDIX

Claim 1: A method for use in managing data in a database system, comprising:
receiving a request to perform an operation on a set of target data residing in the
database;

executing the operation in the database on the set of target data; and
at some point after execution has begun, placing a lock on the target data to
prevent concurrent execution of other operations on the target data.

Claim 2: The method of claim 1, comprising placing an initial lock on the target
data at a level that prevents concurrent execution of at least one operation and, at some
point after execution has begun, placing a final lock on the target data at a level that
prevents concurrent execution of a larger set of operations.

Claim 3: The method of claim 2, where the initial lock allows concurrent
execution of operations that involve reading the target data.

Claim 4: The method of claim 2, where the final lock prevents concurrent
execution of all operations on the target data.

Claim 5: The method of claim 2, further comprising allowing a user to specify
the type of lock initially placed on the data.

Claim 6: The method of claim 1, where the operation is one of the following
types: a COLLECT STATISTICS operation, a CREATE INDEX operation, and an
ALTER TABLE operation.

NCR Docket No. 8779

Claim 7: A database system comprising:
at least one storage device;
at least one computing node configured to deliver data to and retrieve data from the storage device; and
a database-management component configured to:
receive a request to perform an operation on a set of target data residing in the database;
execute the operation in the database on the set of target data; and
at some point after execution as begun, place a lock on the target data to prevent concurrent execution of other operations on the target data.

Claim 8: The system of claim 7, where the database-management system is configured to place an initial lock on the target data at a level that prevents concurrent execution of at least one operation and, at some point after execution has begun, placing a final lock on the target data at a level that prevents concurrent execution of a larger set of operations.

Claim 9: The system of claim 8, where the initial lock allows concurrent execution of at least one other operation on the target data.

Claim 10: The system of claim 8, where the subsequent lock prevents concurrent execution of all other operations on the target data.

Claim 11: The system of claim 8, where the database-management system is configured to allow a user to specify the type of lock initially placed on the data.

Claim 12: The system of claim 7, comprising multiple computing nodes and multiple storage devices, where each storage node is configured to manage storage of data on at least a subset of the storage devices.

NCR Docket No. 8779

Claim 13: The system of claim 12, where the database-management system is configured to place the lock on a block of data that is spread across more than one of the storage devices.

Claim 14: The system of claim 7, where the operation is one of the following types: a COLLECT STATISTICS operation, a CREATE INDEX operation, and an ALTER TABLE operation.

Claim 15: A computer program, stored on at least one computer-readable storage medium, for use in managing data in a database system, comprising executable instructions that, when executed by a computer, cause the computer to:

- receive a request to perform an operation on a set of target data residing in the database;
- execute the operation in the database on the set of target data; and
- at some point after execution has begun, place a lock on the target data to prevent concurrent execution of other operations on the target data.

Claim 16: The program of claim 15, where the program causes the computer to place an initial lock on the target data at a level that prevents concurrent execution of at least one operation and, at some point after execution has begun, placing a final lock on the target data at a level that prevents concurrent execution of a larger set of operations.

Claim 17: The program of claim 16, where the initial lock allows concurrent execution of at least one other operation on the target data.

Claim 18: The program of claim 16, where the subsequent lock prevents concurrent execution of all other operations on the target data.

Claim 19: The program of claim 16, where the program causes the computer to allow a user to specify the type of lock initially placed on the data.

NCR Docket No. 8779

Claim 20: The program of claim 15, where the operation is one of the following types: a COLLECT STATISTICS operation, a CREATE INDEX operation, and an ALTER TABLE operation.

Claim 21: A method for use in managing data in a database system, comprising:
receiving an instruction from a user to perform a data-definition operation on a set of target data residing in the database;
placing an initial lock on the target data at a level that allows at least one concurrent operation on the target data;
executing the operation in the database on the set of target data; and
at some point after execution has begun, placing a final lock on the target data at a level that excludes all other concurrent operations on the target data.

Claim 22: The method of claim 21, where the initial lock excludes at least some concurrent operations on the target data.

Claim 23: The method of claim 21, further comprising allowing a user to select the level of the initial lock.

Claim 24: The method of claim 21, where placing an initial lock on the target data includes placing one of the following types of locks on the target data: an ACCESS lock; a READ lock; and a WRITE lock.

Claim 25: The method of claim 21, where placing a final lock on the target data includes placing an EXCLUSIVE lock on the target data.

Claim 26: The method of claim 21, where placing an initial lock on the target data includes locking an entire table.

NCR Docket No. 8779

Claim 27: The method of claim 21, where receiving the instruction from the user includes receiving an instruction to perform one of the following operations: a CREATE INDEX operation, a COLLECT STATISTICS operation, and an ALTER TABLE operation.

Claim 28: A method for use in managing data in a database system, the method comprising:

- receiving an instruction to perform a MODIFY DATABASE/USER operation on a set of target data;
- initiating execution of the operation; and
- at some point during execution of the operation, concurrently executing another operation on objects within the targeted database or user.

Claim 29: The method of claim 28, further comprising maintaining an ACCESS lock on the target database or user and no locks on the immediate parent of the targeted database or user during execution of the MODIFY DATABASE/USER operation.

Claim 30: A method for use in managing data in a database system, comprising:

- receiving an instruction from a user to perform a data-definition operation on a set of target data;
- placing an initial lock on the target data at a level that prevents at least one type of concurrent operation on the target data;
- initiating execution of the operation on the target data; and
- at some point after execution has begun, placing a final lock on the target data at a level that excludes all types of concurrent operations on the target data.